Amendments to the Specification

Please add the following paragraph after the paragraph beginning at page 5, line 20, with the following new paragraph:

Figure 8 illustrates an exemplary implementation of an LDPC encoder 1000.

Please replace the paragraph beginning at page 5, line 23, with the following rewritten paragraph:

Figure ~~8~~ 9 graphically illustrates the effect of making three copies of the small LDPC graph shown in Fig.3.

Please replace the paragraph beginning at page 5, line 26, with the following rewritten paragraph:

Figure ~~9~~ 10 illustrates the parity check matrix representation of the LDPC graph illustrated in Fig. 8.

Please delete the paragraph beginning at page 5, line 29.

Please replace the paragraph beginning at page 5, line 32, with the following rewritten paragraph:

Figure ~~11~~ 10 illustrates the effect of replacing the 3x3 identity matrices shown in Fig. 9 with cyclic permutation matrices in accordance with one exemplary embodiment of the present invention.

Please replace the paragraph beginning at page 16, line 33, with the following rewritten paragraph:

PAGE 8/32 * RCVD AT 2/17/2005 12:03:25 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/2 * DNIS:8729306 * CSID:17325429071 * DURATION (mm-ss):10-22

BEST AVAILABLE COPY

Figs. ~~6~~ 5 and ~~7 illustrates~~ 6 illustrate the encoding process for the LDPC code shown in Fig. 3. As described earlier, the encoding preprocessing step requires rearranging the rows and columns of the parity check matrix H shown in Fig. 4 into some lower triangular form. One exemplary way of rearrangement is illustrated in Fig. ~~6~~ 5, by swapping row 2 and row 4 in the original matrix.

Please replace the paragraph beginning at page 17, line 20, with the following rewritten paragraph:

Fig. ~~7~~ 6 illustrates the actual encoding process given an information block $s=[1]$ 801 and pre-computed matrices shown in Fig. 6. Standard multiplication of a vector by a matrix allows computation of As 802, $T^{-1}As$ 803, $ET^{-1}As$ 804, $ET^{-1}As+Cs$ 805, $p_1=\phi^{-1}(ET^{-1}As+Cs)$ 806, Bp_1 807, Bp_1+As 808, and $p_2=T^{-1}(Bp_1+As)$ 809. Note that multiplication by T^{-1} is performed using back substitution as described earlier. The final result, the coded bits $x=[p_1,p_2,s]$ are shown in vector 810.

Please replace the paragraph beginning at page 18, line 17, with the following rewritten paragraph:

Fig. ~~8~~ 7 illustrates the encoding process as a sequence of those two simple operations corresponding to the LDPC code shown in Fig. 3. An exemplary memory 902 stores information bits, coded bits, and intermediate variables. In Fig. ~~8~~ 7, location 0 of the memory 902 is assigned to store the single information bit s; location 1 is assigned to store parity bit p_1; locations 2 to 4 are assigned to store parity bits p_2. Additional memory space is provided to hold intermediate values. The exemplary memory 902 provides locations 5 to 7 to store the

PAGE 9/32 * RCVD AT 2/17/2005 12:03:25 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/2 * DNIS:8729306 * CSID:17325429071 * DURATION (mm-ss):10-22

value of As and later that of $Bp_1 + As$; it provides locations 9 to 11 to store $T^{-1}As$; it provides locations 12 to store $ET^{-1}As$.

Please replace the paragraph beginning at page 18, line 29, with the following rewritten paragraph:

With respect to the above allocation of memory 902, the encoding process illustrated in Fig. ~~7~~ 6 as matrix multiplication with vectors is decomposed into a sequence of operations (0 a b) and (1 a b) listed in Table 904. For clarity, table 904 shows the sequence of instructions, one per row, together with their respective matrix multiplication counterparts. For example, multiplication As is decomposed to two instructions: (0 5 0) followed by (0 7 0). Table 906 shows the contents of memory locations 0 through 11 at the time an instruction shown in the corresponding row on table 904 is executed. The result of executing of instruction on table 904 is shown in the next row of table 906. Suppose we encode the same information bits as in Fig. 6 by storing s=[1] into location 0, as illustrated in the first row of Table 906. Operations executing instruction (0 5 0) followed by instruction (0 7 0) gives result As =(1 0 1) in locations from 5 to 7, as shown in row three of block 906. This is the same result as its counterpart in Fig. 6. Table 906 illustrates the complete encoding process in terms of the content of memory locations 0 through 11 as the sequence of elementary instructions in table 904 is executed.

Please replace the paragraph beginning at page 22, line 34, with the following rewritten paragraph:

Let us now consider the introduction of rotations into our example. This can be illustrated by replacing each of the 3x3 identity matrixes shown in Fig. ~~9~~ 10 with 3x3 cyclic

PAGE 10/32 * RCVD AT 2/17/2005 12:03:25 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/2 * DNIS:8729306 * CSID:17325429071 * DURATION (mm-ss):10-22

permutation matrices as shown in Fig. 11. Note that there are three possibilities for the cyclic permutation matrix used in Fig. 11. It is possible to indicate the particular permutation matrix to be substituted for an identity matrix by indicating whether the permutation matrix has a "1" located in the first, second or third position in the first row of the permutation matrix. For example, in the case of matrix 1302, beginning at the top left and proceeding to the bottom right corner the rotations could be specified by the sequence (2, 2, 3, 3, 1, 1, 1, 3, 2, 1, 2, 3).

Please replace the paragraph beginning at page 23, line 30, with the following rewritten paragraph:

We discussed above how to vectorize an encoder ~~900~~ to encode Z parallel copies of the projected graph. By introducing switches into the message paths to perform rotations, we encode the LDPC code defined in Fig. 11.

PAGE 11/32 * RCVD AT 2/17/2005 12:03:25 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/2 * DNIS:8729306 * CSID:17325429071 * DURATION (mm-ss):10-22